

5307.31-m

ORC 68-28
DECEMBER 1968

TWO SENSITIVITY ANALYSIS PROBLEMS IN NETWORKS

by

JOSEPH DEGRAND

AD 685829

**OPERATIONS
RESEARCH
CENTER**

UNIVERSITY OF CALIFORNIA • BERKELEY

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va. 22151



This document has been approved for public release and sale; its distribution is unlimited. The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

TWO SENSITIVITY ANALYSIS
PROBLEMS IN NETWORKS

by

Joseph Degrand
Operations Research Center
University of California, Berkeley

December 1968

ORC 68-28

This research has been supported by the U. S. Army Research Office-Durham under Contract DA-31-124-ARO-D-331 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

ACKNOWLEDGMENTS

I wish to acknowledge the many constructive comments and criticisms of this dissertation by my M.S. Committee. I particularly recognize the fact that my Chairman, Dr. R. B. Potts has made a major contribution to my education in analitical approaches to practical transportation problems.

ABSTRACT

This paper is concerned with two sensitivity analysis problems on a network each arc of which is subject to one or to more than one reduction in its length. Suppose that a total number of s reductions can occur on the whole network. The two problems are the following:

(i) Which arc should be reduced such that the shortest route from source to any node i is reduced the most?

(ii) Which arcs should be reduced such that the sum of all the shortest routes between each pair of nodes is reduced the most?

In (i) we generalize Dantzig's algorithm concerning the shortest route from source to sink in a graph. In (ii) we generalize Dantzig's algorithm concerning the shortest routes between each pair of nodes in a directed graph.

TABLE OF CONTENTS

	PAGE
I. INTRODUCTION	1
II. SHORTEST ROUTE FROM SOURCE TO ANY NODE IN THE GRAPH	2
II.1 Introduction	2
II.2 Formulation of the Problem	2
II.3 Algorithm	4
a) One Improvement Can Occur on Each Arc	4
b) m Improvements Can Occur on Each Arc ($m > 1$)	10
II.4 Application	15
III. SHORTEST ROUTE BETWEEN EACH PAIR OF NODES IN A GRAPH	16
III.1 Definition of the Problem	16
III.2 Algorithm	16
a) Each Arc Can Be Improved Once	17
b) Each Arc Can Be Improved m Times ($m > 1$)	30
III.3 Applications	32
REFERENCES	34
APPENDIX: APPLICATION OF THE ALGORITHM OF SECTION II TO THE MAXIMAL FLOW PROBLEM	A.1

CHAPTER I

INTRODUCTION

The purpose of this paper is to solve several sensitivity analysis problems in networks.

A network is defined by a set of nodes i , a set of arcs (i_u, i_v) and a set of distances $d(i_u, i_v)$. The arcs are subject to one or to several reductions of their length. The problem consists in determining on which arcs the reductions should occur such that the shortest route is reduced the most.

In Part II, we consider this problem for the shortest route from source to any node in a graph. Two cases have to be considered:

- Each arc is subject to only one reduction, the total number of reductions being s .
- Each arc is subject to more than one reduction, say m , the total number of reductions being s .

As an application of Part II, we indicate in the appendix the equivalence of this problem with the following sensitivity analysis on a maximum flow problem: The capacity of which arcs should be reduced such that the maximum flow from source to sink is reduced the most? We have considered three cases:

- The one-commodity flow problem for a planar graph.
- The two-commodity flow problem for an undirected graph.
- The multi-commodity flow problem for a completely planar graph.

Part III deals with the same sensitivity analysis problem corresponding to the sum of all the shortest routes between each pair of nodes in a graph.

For each part an algorithm to solve the problem is given and short applications referring to transportation problems are indicated.

Since a reduction of the length of an arc can be considered as an improvement on this arc, we will use, in the following, these two words indifferently.

CHAPTER II

SHORTEST ROUTE FROM SOURCE TO ANY NODE IN THE GRAPH

II.1 Introduction

Consider a network $\{N, A\}$, N being a set of nodes, A being a set of arcs. To each arc corresponds a length. There is the shortest path problem from source to any node in the graph. Suppose that the distance of each arc (i_u, i_v) can be reduced m times. The improvement problem is the following: On which arcs should a total number of s improvements occur such that the shortest route from the source to a certain node i in the graph is reduced the most? Several papers have already dealt with this problem, especially two by Richard Wollmer (Ref. 8,9). In (8) he considers the case where only one reduction can occur on each arc, and the remaining length is equal to zero. In (9) he considers the same case, but the remaining length can be any nonnegative number.

This paper considers the possibility of more than one reduction on each arc, and the remaining lengths can take any nonnegative value.

Using several special network properties, the following algorithm provides the means for assessing very large numbers of transportation projects and project combinations without going through the usual complete traffic assignment process. It is capable of evaluating a large number of network alternatives in the same amount of time it would take to evaluate a single alternative using standard procedures and still obtain the same information at a comparable level of detail.

II.2 Formulation of the Problem

Let $\{N, A\}$ be a network. N is a finite nonempty set of nodes i . A is a set of ordered pairs (i_u, i_v) of distinct nodes; each such pair is called an arc. Let n be the number of nodes in N and a the number of arcs in A .

One node in the network is distinguished: node S called source. A path in the network from S to node i_w is a sequence of nodes:

$$P(S, i_w) = S, i_1, i_2, \dots, i_{w-1}, i_w.$$

such that

$$(S, i_1), (i_1, i_2), \dots, (i_{w-1}, i_w) \in A.$$

Consider a function d which attributes to each arc (i_u, i_v) a nonnegative number $d(i_u, i_v)$. Depending on the application this number represents the length, the cost, the time or the risk involved in traversing the arc (i_u, i_v) . If the arc (i_u, i_v) does not exist, put $d(i_u, i_v) = \infty$. Whatever it stands for, we call in the following $d(i_u, i_v)$ the length of the arc (i_u, i_v) . The length of a path is defined by:

$$d[P(S, i_w)] = \sum_{t=1}^w d(i_{t-1}, i_t) \text{ with } i_0 = S.$$

Consider now a function b which attributes to each arc (i_u, i_v) a set of nonnegative numbers $b(i_u, i_v)$. Let $b^0(i_u, i_v), b^1(i_u, i_v), \dots, b^m(i_u, i_v)$ be the elements of this set with:

$$b^0(i_u, i_v) = d(i_u, i_v) > b^1(i_u, i_v) > \dots > b^m(i_u, i_v) > 0$$

for all the arcs (i_u, i_v) .

m , the number of possible improvements, is not necessarily the same for each arc. It can take the values $0, 1, \dots, s$ if s is the total number of improvements on the whole network.

When the length of arc (i_u, i_v) passes from $d(i_u, i_v)$ to $b^\alpha(i_u, i_v)$ with $\alpha \in \{1, 2, \dots, m\}$, we say that the arc (i_u, i_v) has been improved or reduced by

the quantity $d(i_u, i_v) - b^{\alpha}(i_u, i_v)$, and the reduced length or the improved length of arc (i_u, i_v) is $b^{\alpha}(i_u, i_v)$.

Suppose a total number of s improvements can occur on the whole network. If an arc is subject to only one improvement, we assume that the reduced length is $b^1(i_u, i_v)$. If an arc is subject to more than one improvement, we assume that the successive improvements are:

$$b^1(i_u, i_v), \dots, b^m(i_u, i_v).$$

The improvement problem is the following: On which arcs should the s improvements occur such that the length of the shortest route from source to a certain node i is reduced the most? The following algorithm provides us with the solution for all the nodes i of the graph.

II.3 Algorithm

a) One improvement can occur on each arc. The total number of improvements is s .

The first part provides us with the reduced length of the shortest path from S to any node i and with the corresponding arc improvements, but not with the shortest paths themselves. The second part will give us these paths.

First Part:

Since each arc is subject to only one improvement, the reduced length is $b^1(i_u, i_v)$ for arc (i_u, i_v) .

The algorithm is based on a double multi-stage decision process solved by dynamic programming methods. It is developed in s successive steps (s being the total number of improvements that can occur on the whole network), Step r providing us with the lengths of the shortest routes from the source to any node i when r improvements occur and giving us the corresponding

improvements. Furthermore, each step is by itself a multi-stage decision process; it is developed in n successive stages (n being the total number of nodes in the graph), each stage giving us the length of the shortest route from the source to a certain node i and the corresponding arc improvements.

The algorithm consists in a generalization of Dantzig's algorithm for the shortest route problem in a network. (Ref. 2)

Notation:

- $d(i_u, i_v)$ = initial length of arc (i_u, i_v) .
- $b^1(i_u, i_v)$ = improved length of arc (i_u, i_v) .
- For each step we define a labelling function which attributes to each node i a nonnegative value representing the length of the shortest path from source to node i . For Step r the labelling function $V_r(i)$ represents the length of the shortest path from source to node i when at most r improvements occur.
- N_r^k = set of nodes labelled in Step r after Stage k .
- \bar{N}_r^k = complementary set of N_r^k .

Algorithm:

Step 0:

Shortest route *without any improvement*. (Dantzig's algorithm.)

- Stage 0: Suppose $V_0(S) = 0$. The function V_0 is, therefore, defined on the set $N_0^0 = \{S\}$.
- Stage 1: Find the node $i(m)$ which is at the shortest distance from the source S . The node $i(m)$ is defined by:

$$V_0[i(m)] = \min_{i_y \in \bar{N}_0^0} [d(S, i_y)] .$$

Define $N_0^1 = N_0^0 \cup i(m)$.

- Stage k: Suppose we have defined the function V_0 on a set N_0^{k-1} of k nodes. Define the node $i(m)$ by:

$$V_0[i(m)] = \min \left\{ \begin{array}{l} V_0(i_x) + d(i_x, i_y) \\ (i_x, i_y) \in \text{cut}^+(N_0^{k-1}, \bar{N}_0^{k-1}) \end{array} \right.$$

Define $N_0^k = N_0^{k-1} \cup i(m)$.

Go to Step 1 when all the nodes are labelled, i.e., when $k = n$.

Step 1:

Shortest route with one improvement.

- Stage 0: Suppose $V_1(S) = 0$ $N_1^0 = \{S\}$.
- Stage 1: Find the node $i(m)$ which is at the shortest improved distance from the source S . Define the node $i(m)$ by:

$$V_1[i(m)] = \min_{i_y \in \bar{N}_1^0} [b^1(S, i_y)] .$$

[†]A cut (X, \bar{X}) is a set of arcs (i_u, i_v) such that $i_u \in X$, $i_y \in \bar{X}$, X and \bar{X} being a partition of the nodes.

Define $N_1^1 = N_1^0 \cup i(m)$.

- Stage k: The function V_1 is defined on a set N_1^{k-1} of k nodes.

Define the node $i(m)$ by:

$$V_1[i(m)] = \min [V_1(i_x) + d(i_x, i_y), V_0(i_x) + b^1(i_x, i_y)]$$

$$\left\{ \begin{array}{l} i_x \in N_1^{k-1} \\ i_y \in \bar{N}_1^{k-1} \\ (i_x, i_y) \in \text{cut } (N_1^{k-1}, \bar{N}_1^{k-1}) \end{array} \right.$$

Define $N_1^k = N_1^{k-1} \cup i(m)$.

Go to Step 2 when $k = n$.

Step r:

Shortest route with r improvements.

- Stage 0: Suppose $V_r(S) = 0$ $N_r^0 = \{S\}$.

- Stage k: The function V_r is defined on the set N_r^{k-1} of k nodes.

Define node $i(m)$ by:

$$V_r[i(m)] = \min [V_r(i_x) + d(i_x, i_y), V_{r-1}(i_x) + b^1(i_x, i_y)]$$

$$\left\{ \begin{array}{l} i_x \in N_r^{k-1} \\ i_y \in \bar{N}_r^{k-1} \\ (i_x, i_y) \in \text{cut } (N_r^{k-1}, \bar{N}_r^{k-1}) \end{array} \right.$$

Define $N_r^k = N_r^{k-1} \cup i(m)$.

When $k = n$ stop if $r = s$. Otherwise go to Step $r + 1$.

Remark:

1) If at any stage the determination of $i(m)$ is not unique, choose arbitrarily one determination.

2) If the arc (i_u, i_v) does not exist $d(i_u, i_v) = \infty$. If $b^1(i_u, i_v)$ exists, i.e., if $b^1(i_u, i_v) \neq \infty$; and if this arc is improved for a certain node, say i_e , that means that we add to the original set of arcs A a new arc (i_u, i_v) with length $b^1(i_u, i_v)$ and that this arc belongs to the shortest route from source to node i_e .

Second Part:

The previous algorithm provides us with the length of the shortest route from source S to any node i and with the corresponding arc improvements. Let us see now how to find the different shortest paths themselves.

Consider Step s , Stage k . Let $P_s[i(m)]$ denote the shortest path from S to node $i(m)$ when s improvements occur. Depending on which arc the last improvement occurs, the definition of $P_s[i(m)]$ is different:

- If $V_s[i(m)] = V_s(i_x) + d[i_x, i(m)]$ $i_x \in N_s^{k-1}$, the last arc of the path $P_s[i(m)]$ is not improved but the improvement occurs all before.

Define $P_s[i(m)] = P_s(i_x) \cup i(m)$.

- If $V_s[i(m)] = V_{s-1}(i_x) + b^1[i_x, i(m)]$ $i_x \in N_s^{k-1}$, the last improvement occurs on arc $[i_x, i(m)]$ and the other improvements occurred before.

Define $P_s[i(m)] = P_{s-1}(i_x) \cup i(m)$.

- If $P_s[i(m)]$ is not determined uniquely, choose arbitrarily one determination.

Each step provides us with a tree. Let us call T_s the tree corresponding to Step s . The tree T_0 gives also the shortest paths from S to any node i .

But for the other trees this is not necessarily true.

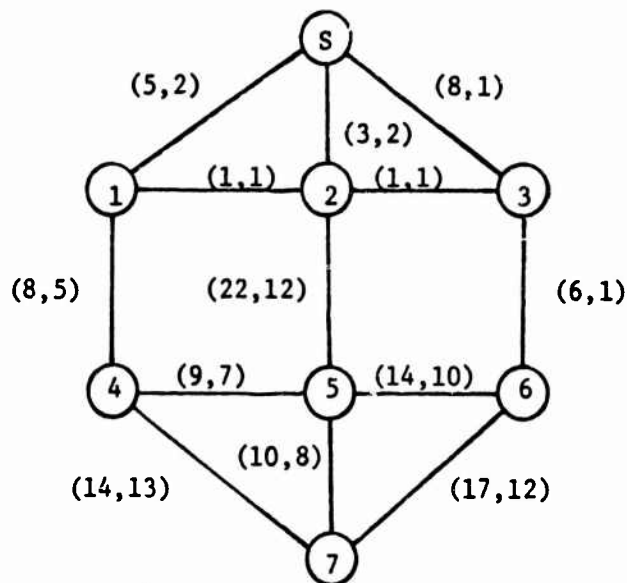
Consider the successive trees T_0, T_1, \dots, T_s . From the first part of the algorithm we know on which arcs the improvements occur. Suppose that the last improvement is on (i_α, i_β) . It belongs necessarily to T_s . In order to find the shortest path $P_s(i)$, trace back from i to i_α on the tree T_s . From i trace back on T_{s-1} until you reach another improved arc, etc... till you reach S .

If we call $P'_r(i_x, i_y)$ the path from i_x to i_y on the tree T_r ; and if the first improvement occurs on arc (i_γ, i_ϵ) , we have:

$$P_s(i) = P_{s-1}(i_\alpha) \cup P'_s(i_\alpha, i) = P_0(i_\gamma) \cup P'(i_\gamma, \dots) \cup \dots \cup P'_{s-1}(\dots, i_\alpha) \cup P'_s(i_\alpha, i).$$

Example:

In order to know at the end on which arc the improvement occurred, we indicate it on each label. For instance, $V_1[i, (i_e, i_f)]$ means: the length of the shortest route from source to node i when one improvement occurs is $V_1(i)$, and the improvement occurs on arc (i_e, i_f) . On each arc we indicate the initial and the reduced distance: $[d(i_u, i_v), b^1(i_u, i_v)]$.



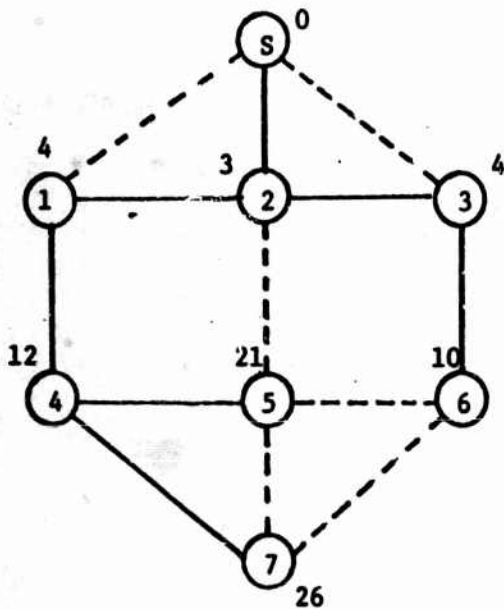


FIGURE 1: No Improvement

$$T_0 = \text{---}$$

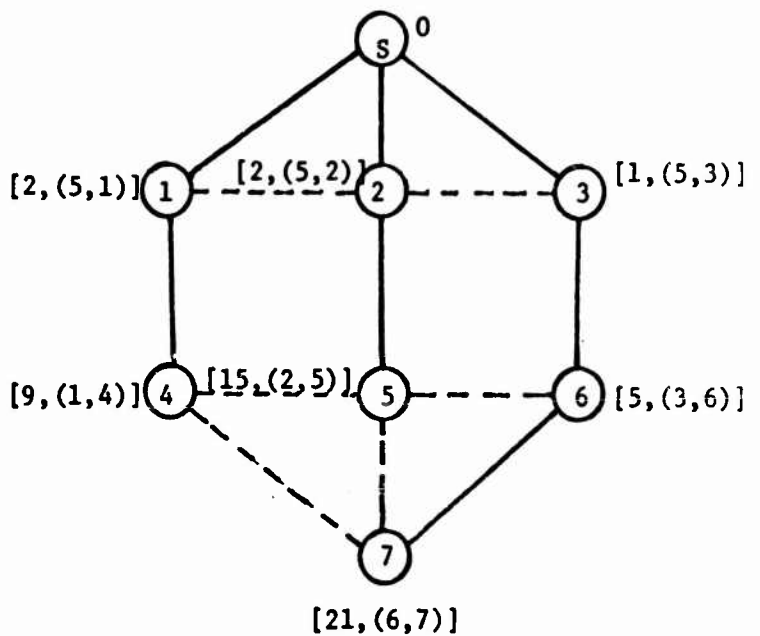


FIGURE 2: One Improvement

$$T_1 = \text{---}$$

Let us find now the shortest paths: For node 5 the improvement occurs on the arc $(2,5)$. Hence, follow the tree T_1 till node 2 and then T_0 till S. So $P_1(5) = S, 2, 5$.

For node 7 the improvement is on the arc $(6,7)$. Follow T_1 till node 6, then follow T_0 till S. So $P_1(7) = P_0(6) \cup 7 = S, 2, 3, 6, 7$.

b) m improvements can occur on each arc ($m > 1$). The total number of improvements is s .

First Part:

The first improvement on arc (i_u, i_v) leads to the reduced length $b^1(i_u, i_v)$. The m -th improvement leads to the reduced length $b^m(i_u, i_v)$. As previously, the algorithm is developed in s successive steps, Step r providing us with the minimal paths corresponding to a total number of most r improvements and giving

us the arcs on which these improvement occur.

Notation:

The same as in Part a. $b^1(i_u, i_v), \dots, b^m(i_u, i_v)$ are the successive improved lengths of arc (i_u, i_v) .

Algorithm:

Step 0:

Shortest route *without any improvement*. Same as in Section a.

Step 1:

Shortest route *with one improvement*. Same as in Section a.

Step r:

$(r > 1)$ shortest route *with r improvements*.

- Stage 0: Suppose $V_r(S) = 0$ $N_r^0 = \{S\}$

- Stage k: The function V_r is defined on the set N_r^{k-1} of k nodes.

Define $i(m)$ by: if $r < m$:

$$V_r[i(m)] = \min \left[V_r(i_x) + d(i_x, i_y), V_{r-1}(i_x) + b^1(i_x, i_y), \dots, \right. \\ \left. \begin{cases} i_x \in N_r^{k-1} & V_o(i_x) + b^r(i_x, i_y) \\ i_y \in \bar{N}_r^{k-1} \\ (i_x, i_y) \in \text{cut } (N_r^{k-1}, \bar{N}_r^{k-1}) \end{cases} \right]$$

if $r \geq m$:

$$V_r[i(m)] = \min \left[V_r(i_x) + d(i_x, i_y), V_{r-1}(i_x) + b^1(i_x, i_y), \dots, \right. \\ \left. \begin{cases} i_x \in N_r^{k-1} & V_{r-m}(i_x) + b^m(i_x, i_y) \\ i_y \in \bar{N}_r^{k-1} \\ (i_x, i_y) \in \text{cut } (N_r^{k-1}, \bar{N}_r^{k-1}) \end{cases} \right]$$

or condensed in one expression:

$$V_r[i(m)] = \min \left[V_r(i_x) + d(i_x, i_y), V_{r-1}(i_x) + b^1(i_x, i_y), \dots, \right. \\ \left. \begin{cases} i_x \in N_r^{k-1} & V_{r-\min(r,m)}(i_x) + b^{\min(r,m)}(i_x, i_y) \\ i_y \in \bar{N}_r^{k-1} \\ (i_x, i_y) \in \text{cut } (N_r^{k-1}, \bar{N}_r^{k-1}) \end{cases} \right]$$

Second Part:

In the same way as in Section a we determine the shortest path from S to i . But now more than one improvement can occur on each arc. Thus if:

$$V_s[i(m)] = V_{s-t}(i_x) + b^t[i_x, i(m)] \quad i_x \in N_s^{k-1}, \quad t \leq \min(s, m)$$

t improvements occur on arc $[i_x, i(m)]$ and $(s-t)$ occurred before. Define

$P_s[i(m)] = P_{s-t}(i_x) \cup i(m)$. Consider now the successive trees

T_0, T_1, \dots, T_s . Suppose that the last improvement occurred on (i_α, i_β) and

that there was only one improvement on that arc. It belongs necessarily to T_s .

As in Section a, in order to find the shortest path, trace back from i to i_α

on the tree T_s . From i trace back on T_{s-1} until you reach another improved

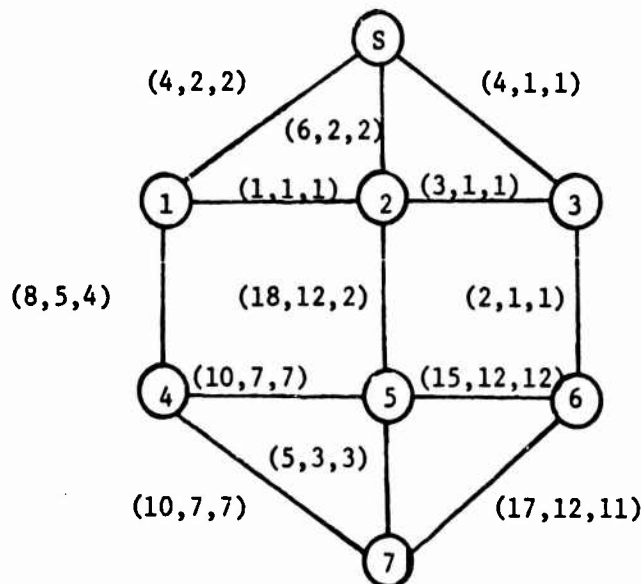
arc, etc. till you reach S .

More generally, suppose we follow the tree T_q ($0 < q \leq s$) ; and we encounter an arc improved α times, say arc (i_β, i_ρ) , ($\alpha \leq \min(q, m)$) . Add this arc to $P_s(i)$ and switch to the tree $T_{q-\alpha}$. Suppose the first improvement is on arc (i_γ, i_ϵ) and that there is only one improvement on this arc. If we call $P'_r(i_x, i_y)$ the path from i_x to i_y on the tree T_r , we have:

$$P_s(i) = P_0(i_\gamma) \cup P'_1(i_\gamma, \dots) \cup \dots \cup P'_{q-\alpha}(\dots, i_\beta) \cup P'_q(i_\beta, \dots) \\ \cup \dots \cup P'_{s-1}(\dots, i_\alpha) \cup P'_s(i_\alpha, i) .$$

Example:

$s = m = 2$. In order to know at the end on which arcs the improvements occurred, we indicate it on each label. $V_2[i, (i_e, i_f), (i_g, i_h)]$ means that the length of the shortest route from source to node i , when two improvements occur is $V_2(i)$ and that one improvement occurs on arc (i_e, i_f) and the other on arc (i_g, i_h) . $V_2[i(i_e, i_f)^2]$ means that both improvements occur on arc (i_e, i_f) . On each arc we indicate the initial and the improved lengths: $[d(i_u, i_v), b^1(i_u, i_v), b^2(i_u, i_v)]$.



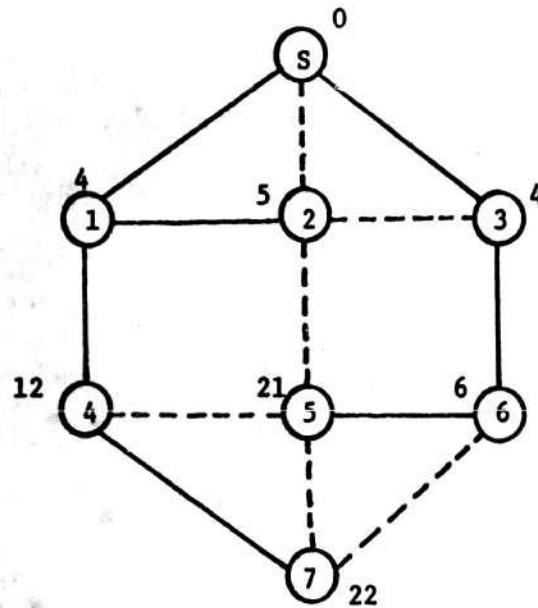


FIGURE 1: No Improvement

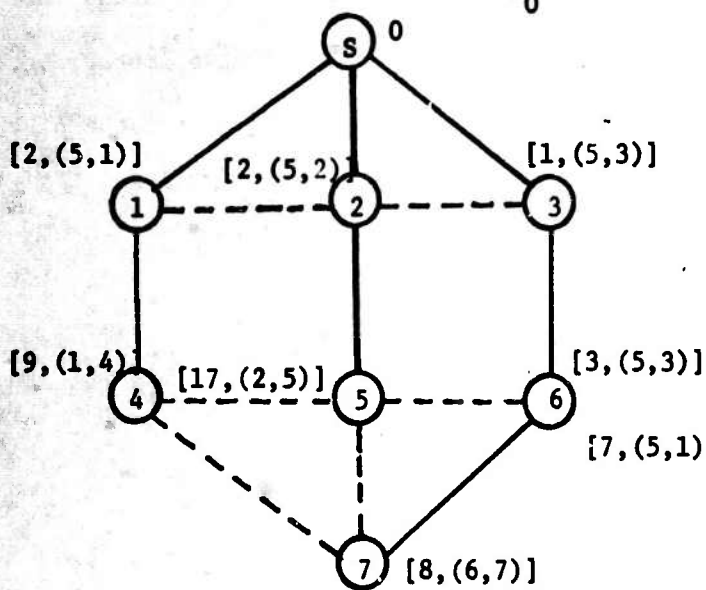
 $T_0 = \text{---}$ 

FIGURE 2: One Improvement

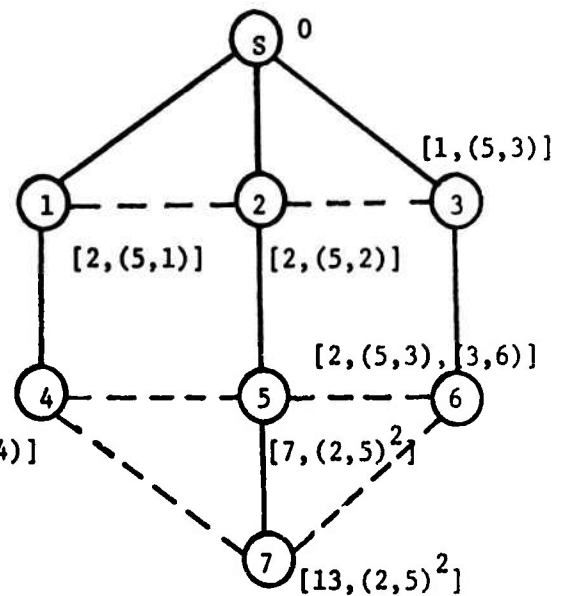
 $T_1 = \text{---}$ 

FIGURE 3: Two improvements

 $T_2 = \text{---}$

The shortest path from S to 7 is:

$$P_2(7) = P_0(2) \cup P'_2(2,7) = (S,1,2) \cup (5,7) = S, 1, 2, 5, 7.$$

The two improvements occur on arc (2,5).

II.4 Application

Consider a transportation network between two towns. A certain amount of money is available to improve the network, i.e., to decrease the distance or the travel time of several arcs. The improvement problem can be formulated like this: How should this money be spent or which arc lengths should be reduced such that the shortest route between the two towns is reduced the most?

CHAPTER III

SHORTEST ROUTE BETWEEN EACH PAIR OF NODES IN A GRAPH

III.1 Definition of the Problem

The length of the shortest route is sought between every pair of nodes in a directed graph (N, A) . Let $\{1, 2, \dots, n\}$ be the set of nodes in the graph and let a be the number of arcs. To each arc corresponds a set of nonnegative numbers: $d(i, j) > b^1(i, j) > \dots > b^m(i, j) > 0$. $d(i, j)$ is the initial length of arc (i, j) if it exists. $b^1(i, j), \dots, b^m(i, j)$ are the lengths after the consecutive improvements in this order. If the arc (i, j) does not exist, put $d(i, j) = \infty$. Put $d(i, i) = 0$ for all i . The fixed number m of possible improvements is not necessarily the same for each arc.

Suppose s improvements can occur on the whole network? The improvement problem is the following: Which arcs should be improved such that the sum of all the shortest routes between each pair of nodes is reduced the most?

III.2 Algorithm

Dantzig's algorithm to find the length of all the shortest routes between each pair of nodes in a directed graph is well known. (Ref. 3.) Normally, to find the best improvements, we should apply Dantzig's algorithm as many times as there are possible combinations of improvements. For example, if r is the number of arcs and if s is greater than m for all the arcs, we would have $\binom{s+r-1}{s}$ possible solutions to compare. Obviously, this leads to too many computations.

In order to reduce the number of matrices to be computed, we consider the consecutive nodes in the following order: Take node 1. Then take the next

node in the set $\{2, \dots, n\}$ such that no loop[†] is created between this node and node 1. Go on in avoiding as long as possible to form a loop. If it is no longer possible to avoid it and if there is a choice between two nodes, take the node which creates the smallest number of loops. Go on until all the nodes have been considered.

The way in which the algorithm works will show the obvious advantage of this procedure.

We consider two cases:

- Each arc can be improved once.
- Each arc can be improved m times. ($m > 1$).

a) Each arc can be improved once. The total number of improvements is s . $b^1(i,j)$ is the improved distance of arc (i,j) .

Notation:

- a $p \times p$ matrix is a square matrix with p columns and p rows.
- $((i,j))^{i_\alpha, j_\beta}$ is the matrix corresponding to the improvement on arc (i_α, j_β) . This improvement is obtained by finding the shortest improved route from i to j .
- $((a^{r,n}(i,j)))$ is the matrix of the lengths of the shortest routes between each pair of nodes when the r improvements occur which make the sum of these lengths the shortest.
- $a^{r,k}(i,j)$ is the length of the shortest route from i to j at Stage k when r improvements occur.

[†] A loop is a sequence of arcs (u_1, u_2, \dots, u_t) , whatever the direction of these arcs may be, each arc u_k being connected to arc u_{k-1} by one of its extremities and to u_{k+1} by the other. Furthermore u_1 and u_t join at the same node.

- $\alpha^{r,k}(i,j)$ is the set of the lengths of the shortest routes between i and j corresponding to the new improvements introduced by Stage k when a total number of r improvements occur.
- $\beta^{r,k}(i,j)$ is the set of the lengths of the shortest routes between i and j corresponding to each improvement considered in Stage k when a total number of r improvements occur.
- N^k is the set of k nodes for which the problem is solved.
- \bar{N}^k is the complementary set of N^k .

Algorithm:

The algorithm is developed in s steps, Step r providing us with the solution corresponding to r improvements. Suppose that the consecutive nodes are in the following order: $1, 2, \dots, n$.

Step 0:

Find the matrix $((a^{0,n}(i,j)))$ of the lengths of the shortest routes between each pair of nodes *without any improvement*. (Dantzig's algorithm.)

- Stage 1: $N^1 = \{1\}$
- Stage 2: $N^2 = \{1, 2\}$
 $a^{0,2}(1,2) = d(1,2) \quad a^{0,2}(2,1) = d(2,1)$
- Stage k : $N^k = \{1, 2, \dots, k\}$

Assume that for the first $k-1$ nodes $1, \dots, k-1$ the optimal distances $a^{0,k-1}(i,j)$ are known. These nodes form the set N^{k-1} . Add a node k and find the optimal distances for the nodes $1, \dots, k$.

For $i = 1, \dots, k - 1$ the shortest route from k to i is:

$a^{0,k}(k,i) = \min[d(k,j) + a^{0,k-1}(j,i)] \quad j = 1, \dots, k - 1$. The minimal route from k to i starts with some arc (k,j) followed by a minimal route from j to i that does not go through k . The minimum of these alternatives is the solution.

$$a^{0,k}(i,k) = \min[a^{0,k-1}(i,j) + d(j,k)] \quad j = 1, \dots, k - 1.$$

The minimal route ends with an arc (j,k) preceded by a minimal route from i to j that does not go through k . The minimum of these alternatives is the solution.

For $i = 1, \dots, k - 1$ and $j = 1, \dots, k - 1$

$$a^{0,k}(i,j) = \min[a^{0,k-1}(i,j), a^{0,k}(i,k) + a^{0,k}(k,j)].$$

The entries in the $(k - 1) \times (k - 1)$ matrix $a^{0,k-1}(i,j)$ are replaced by $[a^{0,k}(i,k) + a^{0,k}(k,j)]$ if the latter sum is smaller. If $k = n$ go to Step 1. Otherwise, go to stage $k + 1$.

Step 1:

Find the matrix $((a^{1,n}(i,j)))$ of all the lengths of the shortest routes with one length improvement such that the sum of all these lengths is the smallest. We consider the consecutive nodes in the same order as in Step 0.

- Stage 1: $N^1 = \{1\}$.

- Stage 2: $N^2 = \{1, 2\}$. Consider the improved length from node 1 to node 2. It is $b^1(1, 2)$. The improved length from 2 to 1 is $b^1(2, 1)$. Compute the two corresponding matrices:

$$((1,2))^{1,2} = \begin{pmatrix} 0 & b^1(1,2) \\ d(2,1) & 0 \end{pmatrix} \quad ((2,1))^{2,1} = \begin{pmatrix} 0 & d(1,2) \\ b^1(2,1) & 0 \end{pmatrix}.$$

- Stage 3: $N^3 = \{1,2,3\}$. Find the shortest route from 1 to 3. We have:

$$\begin{aligned} a^{1,n}(1,3) &= \min[a^{0,n}(1,2) + b^1(2,3) ; b^1(1,3) ; a^{1,n}(1,2) + d(2,3)] \\ &= \min[d(1,2) + b^1(2,3) ; b^1(1,3) ; b^1(1,2) + d(2,3)] . \end{aligned}$$

Suppose the best improvement is on arc (i_1, j_3) . (i_1, j_3) is either $(1,2)$ or $(2,3)$. Compute the matrices corresponding to this arc improvement. For instance, if (i_1, j_3) is $(2,3)$, this matrix $((1,3))^{2,3}$ is computed with the arc lengths:

$$\begin{aligned} d(1,1) &= 0, d(1,2), d(1,3) \\ d(2,1) &, d(2,2) = 0, b^1(2,3) \\ d(3,1) &, d(3,2), d(3,3) = 0. \end{aligned}$$

We do the same for the shortest routes from 2 to 3, from 3 to 1 and from 3 to 2. So we get the new improvements on the arcs (i_2, j_3) , (i_3, j_1) , and (i_3, j_2) . For each new improvement, we have to compute the matrix with the distances $d(i,j)$ except $b^1(i,j)$ for the arc on which the improvement occurs. If the improvements of Stage 2 (improvement on arc $(1,2)$ and on arc $(2,1)$) belong to the set of the new improvements, we are done. Go to Stage 4. If not, compare all the elements of the set $\alpha^{1,3}(1,2)$ with $b^1(1,2)$. If one of these elements is smaller than $b^1(1,2)$, we are done. Go to Stage 4.

If they are all greater than $b^1(1,2)$, we have to compute the 3×3 matrix corresponding to the improvement on arc $(1,2)$.

Do the same for the shortest route from 2 to 1.

- Stage k : We know all the $(k-1) \times (k-1)$ matrices of stage $k-1$.

Compute all the shortest routes with one improvement from every node $i \in N^{k-1}$ to node k and from node k to every node $i \in N^{k-1}$.

For $i = 1, \dots, k-1$ the shortest route from i to k is:

$$a^{1,k}(i,k) = \min[\min \text{ out of } \beta^{1,k-1}(i,j) + d(j,k), \\ a^{0,k-1}(i,j) + b^1(j,k)] \quad j = 1, \dots, k-1$$

$\min \text{ out of } \beta^{1,k-1}(i,j)$ means the smallest element of the set $\beta^{1,k-1}(i,j)$. The shortest route from k to i is:

$$a^{1,k}(k,i) = \min[d(k,j) + \min \text{ out of } \beta^{1,k-1}(j,i), \\ b^1(k,j) + a^{0,k-1}(j,i)] \quad j = 1, \dots, k-1.$$

Compute the matrices corresponding to the new improvements. For each improvement, we have a different matrix. Depending on whether an improvement was considered or not previously the formulas to find the shortest route from i to j ($i = 1, \dots, k-1$ and $j = 1, \dots, k-1$) are slightly different. For an improvement which was never considered before the length of the shortest route is:

$$a^{1,k}(i,j) = \min[a^{0,k-1}(i,j), a^{1,k}(i,k) + a^{1,k}(k,j)].$$

For an improvement which was already considered before in one or more of the previous steps the element (i,j) of the corresponding matrix is:

$$a^{1,k}(i,j) = \min[a^{1,k-1}(i,j) , a^{1,k}(i,k) + a^{1,k}(k,j)]$$

$a^{1,k-1}(i,j)$ is the element of the matrix at stage $k - 1$ corresponding to the same improvement we consider now.

If all the improvements of stage $k - 1$ belong to the set of the new improvements, we are done. If one does not belong to this set, say the improvement on arc (i_ϕ, j_ψ) , consider the pair of nodes, say i_z, j_t , for which this improvement gave the shortest distance ρ in stage $k - 1$. If any element of the set $a^{1,k}(i_z, j_t)$ is smaller than ρ , we are done. We do not have to consider any more the improvement on arc (i_ϕ, j_ψ) . If not, we must compute the $k \times k$ matrix corresponding to this improvement. This comparison has only to be done after the first loop has been formed. Before we must consider necessarily at each stage the improvements of the previous stages. If $k = n$ go to the next step. Otherwise, go to stage $k + 1$.

Step r:

Find the matrix $((a^{r,n}(i,j)))$ of all the lengths of all the shortest routes with r length improvements such that the sum of all these lengths is the smallest.

We consider the consecutive nodes in the same order as in Step 0 and 1. The procedure is exactly the same as in Step 1. Let us write the formulas for Stage k :

For $i = 1, \dots, k - 1$

$$a^{r,k}(i,k) = \min[\min \text{ out of } \beta^{r,k-1}(i,j) + d(j,k) ,$$

$$\min \text{ out of } \beta^{r-1,k-1}(i,j) + b^1(j,k)]$$

$$j = 1, \dots, k - 1$$

$$a^{r,k}(k,i) = \min[d(k,j) + \min \text{ out of } \beta^{r,k-1}(j,i) ,$$

$$b^1(k,j) + \min \text{ out of } \beta^{r-1,k-1}(j,i)] \quad j = 1, \dots, k-1$$

For $i = 1, \dots, k-1$ and $j = 1, \dots, k-1$.

We have to consider two different cases. If the new improvement corresponding to the matrix we compute has never been considered before, we have:

$$a^{r,k}(i,j) = \min[a^{r-1,k-1}(i,j) , a^{r,k}(i,k) + a^{r,k}(k,j)]$$

$a^{r-1,k-1}(i,j)$ is the element of the matrix at step $r-1$, stage $k-1$ which corresponds to the same improvements as the $r-1$ first improvements we consider now.

For an improvement which has already been considered in one of the previous stages, the element $a^{r,k}(i,j)$ of the corresponding matrix is:

$$a^{r,k}(i,j) = \min[a^{r,k-1}(i,j) , a^{r,k}(i,k) + a^{r,k}(k,j)]$$

$a^{r,k-1}(i,j)$ is the element of the matrix at Step r , stage $k-1$ which corresponds to the same improvements we consider now.

We have to make the same comparisons as in Step 1 in order to determine the matrices we must compute. If $r = s$, make the sum of the elements of the different matrices. The improvements which correspond to the matrix that gives the smallest sum are the solution of our problem. If $r < s$ go to step $r+1$.

Conclusion:

The way the algorithm works makes the advantages of the special choices of the consecutive nodes clearer. If by adding node k to the set N^{k-1} in passing from stage $k-1$ to Stage k , we do not form any loop, only one improvement is added to the previous improvements. It is the improvement on the

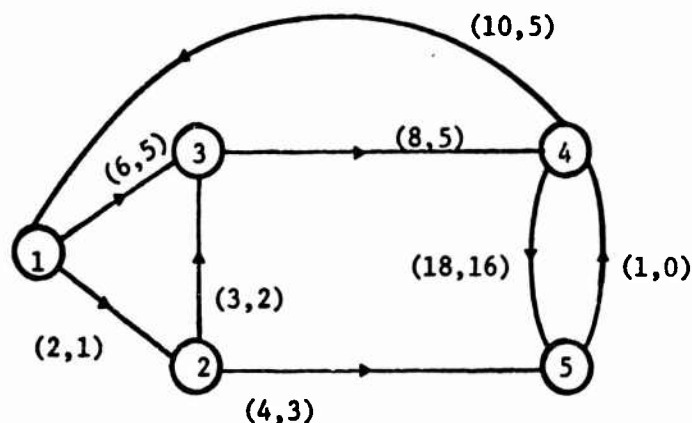
arc p which joins node k to the set of the $k - 1$ first nodes. So the number of matrices to be computed increases only by one unit at this stage. But if we create a loop in adding node k to the first $k - 1$ nodes, two new improvements have to be considered, the first on the arc p and the second on the arc closing the loop. Hence, by this procedure the least possible number of matrices is added at each stage.

For instance, in the following numerical example, we have to compute one matrix at Stage 2, two matrices at Stage 3, three matrices at Stage 4 and six matrices at Stage 5.

Remark:

If there is a tie in the relation (r,k) on Page 22, carry on the improvement from stage $(k - 1)$ and do not consider the new improvement.

Example:



The consecutive nodes are: 1 , 2 , 5 , 3 , 4

Step 0:

- Stage 1: $N^1 = \{1\}$

- Stage 2: $N^2 = \{1, 2\}$. The distance matrix is

$$\begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} 0 & 2 \\ \infty & 0 \end{bmatrix} \end{array}$$

- Stage 3: $N^3 = \{1, 2, 5\}$. The distance matrix is

$$\begin{array}{ccc} & \begin{array}{ccc} 1 & 2 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 5 \end{array} & \begin{bmatrix} 0 & 2 & 6 \\ \infty & 0 & 4 \\ \infty & \infty & 0 \end{bmatrix} \end{array}$$

- Stage 4: $N^4 = \{1, 2, 5, 3\}$. The distance matrix is

$$\begin{array}{cccc} & \begin{array}{cccc} 1 & 2 & 5 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 5 \\ 3 \end{array} & \begin{bmatrix} 0 & 2 & 6 & 5 \\ \infty & 0 & 4 & 3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{bmatrix} \end{array}$$

- Stage 5: $N^5 = \{1, 2, 5, 3, 4\}$. The distance matrix is

$$\begin{array}{ccccc} & \begin{array}{ccccc} 1 & 2 & 5 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 5 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 2 & 6 & 5 & 7 \\ 15 & 0 & 4 & 3 & 5 \\ 11 & 13 & 0 & 16 & 1 \\ 18 & 20 & 24 & 0 & 8 \\ 10 & 12 & 16 & 15 & 0 \end{bmatrix} \end{array}$$

Step 1:

- Stage 1: $N^1 = \{1\}$

- Stage 2: $N^2 = \{1, 2\}$

$$((1, 2))^{1, 2} = \begin{bmatrix} 0 & 1 \\ \infty & 0 \end{bmatrix}$$

- Stage 3: $N^3 = \{1, 2, 5\}$. The shortest route from 1 to 5 is

$$\min \left(\begin{array}{cc} 1 + 4 & , & 2 + 3 \\ \text{improvement} & & \text{improvement} \\ \text{on arc } (1, 2) & & \text{on arc } (2, 5) \end{array} \right) = 5$$

There is a tie. Keep the improvement on (1,2). The length of the shortest route from 2 to 5 is 3. The corresponding improvement is on (2,5). The distances from 5 to 1 and from 5 to 2 remain ∞ . The set of the new improvements is (1,2), (2,5). Hence, we have to consider the two matrices:

$$(((1, 2), (1, 5)))^{1, 2} = \begin{array}{c} \begin{matrix} & 1 & 2 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \begin{bmatrix} 0 & 1 & 5 \\ \infty & 0 & 4 \\ \infty & \infty & 0 \end{bmatrix} \end{array}$$

$$((2, 5))^{2, 5} = \begin{array}{c} \begin{matrix} & 1 & 2 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 5 \end{matrix} \begin{bmatrix} 0 & 2 & 5 \\ \infty & 0 & 3 \\ \infty & \infty & 0 \end{bmatrix} \end{array}$$

- Stage 4: $N^4 = \{1, 2, 5, 3\}$. The length of the shortest route from 1 to 3 is

$$\min \begin{pmatrix} 1+3 & 2+2 & 5 \\ \text{improv.} & \text{improv.} & \text{improv.} \\ \text{on } (1,2) & \text{on } (2,3) & \text{on } (1,3) \end{pmatrix} = 4$$

Keep the previous improvement on arc (1,2). The length of the shortest route from 2 to 3 is 2. The improvement is on (2,3). The distances from 5 to 3, from 3 to 1, from 3 to 2 and from 3 to 5 remain ∞ . The set of the new improvements is: (1,2), (2,3). The corresponding matrices are:

$$(((1,2), (1,5), (1,3)))^{1,2} = \begin{matrix} & \begin{matrix} 1 & 2 & 5 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 5 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 1 & 5 & 4 \\ \infty & 0 & 4 & 3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{bmatrix} \end{matrix}$$

$$((2,3))^{2,3} = \begin{matrix} & \begin{matrix} 1 & 2 & 5 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 5 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 2 & 6 & 4 \\ \infty & 0 & 4 & 2 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{bmatrix} \end{matrix}$$

The improvement on arc (2,5) of Stage 3 is not among the new improvements. It was found in determining the shortest route from 2 to 5. So we have to compare the old distance from 2 to 5 which is equal to 3 to the elements of $\alpha^{1,4}(2,5)$, i.e., 4 and 4. Since the old distance is shorter than the new ones, we have to consider the improvement on arc (2,5) and compute the corresponding matrix.

$$((2,5))^{2,5} = \begin{matrix} & \begin{matrix} 1 & 2 & 5 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 5 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 2 & 5 & 5 \\ \infty & 0 & 3 & 3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{bmatrix} \end{matrix}$$

- Stage 5: $N^5 = \{1, 2, 5, 3, 4\}$. The length of the shortest route from 1 to 4 is

$$\min(4 + 8, 5 + 5, 5 + 1, 6 + 0) = 6.$$

The corresponding improvement is on (1,2).

The shortest distance from 2 to 4 is 4. The improvement is on (2,3).

The shortest distance from 5 to 4 is 0. The improvement is on (5,4).

The shortest distance from 3 to 4 is 5. The improvement is on (3,4).

The shortest distance from 4 to 1 is 5. The improvement is on (4,1).

The shortest distance from 4 to 2 is 7. The improvement is on (4,1).

The shortest distance from 4 to 5 is 11. The improvement is on (4,1).

The shortest distance from 4 to 3 is 11. The improvement is on (4,1).

The set of the new improvements is: (1,2), (2,3), (5,4), (3,4), (4,1).

The corresponding matrices are:

$$((5,4))^{5,4} = \begin{matrix} & \begin{matrix} 1 & 2 & 5 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 5 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 2 & 6 & 5 & 6 \\ 14 & 0 & 4 & 3 & 4 \\ 10 & 12 & 0 & 15 & 0 \\ 18 & 20 & 24 & 0 & 8 \\ 10 & 12 & 16 & 15 & 0 \end{bmatrix} \end{matrix}$$

$$\left(\begin{array}{l} ((4,1), (4,2), (4,5), (4,3), \\ (5,1), (5,2), (5,3), (3,1), \\ (3,2), (3,5), (2,1)) \end{array} \right)^{4,1} =$$

	1	2	5	3	4
1	0	2	6	5	7
2	10	0	4	3	5
5	6	8	0	11	1
3	13	15	19	0	8
4	5	7	11	10	0

$$(((1,2), (1,5), (1,3)))^{1,2} =$$

	1	2	5	3	4
1	0	1	5	4	6
2	15	0	4	3	5
5	11	12	0	15	1
3	18	19	23	0	8
4	10	11	15	14	0

$$((3,4))^{3,4} =$$

	1	2	5	3	4
1	0	2	6	5	7
2	15	0	4	3	5
5	11	13	0	16	1
3	15	17	21	0	5
4	10	12	16	15	0

$$(((2,3), (2,4)))^{2,3} =$$

	1	2	5	3	4
1	0	2	6	4	7
2	15	0	4	2	5
5	11	13	0	15	1
3	18	20	24	0	8
4	10	12	16	14	0

In all these matrices, the shortest distance from 2 to 5 is 4. This is greater than the distance found in improving arc (2,5). Hence, we have to compute also the 5x5 matrix corresponding to this improvement.

$$((2,5))^{2,5} = \begin{matrix} & \begin{matrix} 1 & 2 & 5 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 5 \\ 3 \\ 4 \end{matrix} & \left[\begin{array}{ccccc} 0 & 2 & 5 & 5 & 6 \\ 14 & 0 & 3 & 3 & 4 \\ 11 & 13 & 0 & 16 & 1 \\ 18 & 20 & 23 & 0 & 8 \\ 10 & 12 & 15 & 15 & 0 \end{array} \right] \end{matrix}$$

Let us find the sum of all the elements of these matrices. They are in the same order as the matrices have been written: 204, 156, 200, 199, 207, 204. The smallest sum is 156. That corresponds to the improvement on arc (4,1) which gives the solution to our problem.

b) Each arc can be improved m times ($m > 1$) . m is not necessarily the same for each arc.

The Steps 0 and 1 are exactly the same as in Case a. Consider Step r , Stage k . In applying the same rules as previously, we have the formulas:

For $i = 1, \dots, k-1$

$$\begin{aligned} \text{if } r < m \quad a^{r,k}(i,k) = \min[& \min \text{ out of } \beta^{r,k-1}(i,j) + d(j,k) , \\ & \min \text{ out of } \beta^{r-1,k-1}(i,j) + b^1(j,k) , \\ & \cdot \\ & \cdot \\ & a^{0,k-1}(i,j) + b^r(j,k)] \quad j = 1, \dots, k-1 \end{aligned}$$

m is the number of possible improvements on arc (j,k) .

$$\begin{aligned} a^{r,k}(k,i) = \min[& d(k,j) + \min \text{ out of } \beta^{r,k-1}(j,i) , \\ & b^1(k,j) + \min \text{ out of } \beta^{r-1,k-1}(j,i) , \\ & \cdot \\ & \cdot \\ & b^r(k,j) + a^{0,k-1}(j,i)] \quad j = 1, \dots, k-1 \end{aligned}$$

m is the number of possible improvements
on arc (k,j) .

$$\begin{aligned} \text{if } r \geq m \quad a^{r,k}(i,k) &= \min[\min \text{ out of } \beta^{r,k-1}(i,j) + d(j,k) , \\ &\quad \cdot \\ &\quad \cdot \\ &\quad \cdot \\ &\quad \min \text{ out of } \beta^{r,k-1}(i,j) + b^m(j,k)] \\ &\quad j = 1, \dots, k-1 \end{aligned}$$

m is the number of possible improvements
on arc (j,k) .

$$\begin{aligned} a^{r,k}(k,i) &= \min[d(k,j) + \min \text{ out of } \beta^{r,k-1}(j,i) , \\ &\quad \cdot \\ &\quad \cdot \\ &\quad \cdot \\ &\quad b^m(k,j) + \min \text{ out of } \beta^{r-m,k-1}(j,i)] \quad j = 1, \dots, k-1 \end{aligned}$$

The new improvements being known, now we have to compute the corresponding matrices.

Consider the element $a^{r,k}(i,j)$. Suppose that α new improvements have been found either on arc (j,k) or on arc (k,j) . Then $a^{r,k}(i,j) = \min[a^{r-\alpha,k-1}(i,j), a^{r,k}(i,k) + a^{r,k}(k,j)]$ $a^{r-\alpha,k-1}(i,j)$ is the entry (i,j) of the matrix at step $r - \alpha$, stage $k - 1$ which corresponds to the same improvements as the $r - \alpha$ first improvements we consider now.

If no new improvement is neither on arc (k,j) nor on arc (j,k) , we have:

$$a^{r,k}(i,j) = \min[a^{r,k-1}(i,j), a^{r,k}(i,k) + a^{r,k}(k,j)]$$

$a^{r,k-1}(i,j)$ is the element of the matrix at Step r , stage $k - 1$ which corresponds to the same improvements we consider now.

In order to know the matrices we have to compute, we must make the same comparisons as in Step 1 of Case a. If $r = s$ stop after this step. Otherwise go to step $r + 1$.

III.3 Applications

Consider a network of post-delivery by trains or trucks. In order to reduce the travel time between two cities, we can introduce a plane. Between which cities should this plane be used such that the total delivery time is reduced the most?

How to introduce toll on highways? Consider a communication network which exists already or which has to be constructed. In order to pay the construction, we want to introduce taxes on certain arcs. Suppose that an amount of $\alpha(i,j)$ toll on arc (i,j) is equivalent for the user with an increase of $\beta(i,j)$ miles of the length of the arc (i,j) . The problem is the following: On which arcs should the taxes be introduced such that the sum of all the shortest routes is increased the least?

Different hypotheses can be made:

α) The total number of arcs on which we can introduce toll is fixed, say s . The algorithm is the same as in Section a, but now we have to consider increases of the lengths of the arcs instead of decreases.

β) The total amount of taxes to be collected is fixed a priori, say M . For instance, M may be the equivalent of a certain percentage of the sum of all the shortest routes between each pair of nodes in the network. The algorithm is the same as in α), but now we stop at Step r if the sum of the r first increases is greater or equal to M and the sum of the $r - 1$ first increases is less than M .

γ) In α) and β) we minimized the sum of all the travel times between each pair of nodes, but we did not take into account the amount of flow $f(i,j)$ on each arc. Define now the travel time on each arc (i,j) as the product of the number of cars using this arc in a certain period (for instance, in 24 hours),

and the average travel time of a car on this arc. When the total amount of toll is fixed a priori the algorithm is the same as in β) but now we consider new distance-matrices the elements of which are $f(i,j) \times d(i,j)$.

∂) Suppose now that the travel time on each arc is a known increasing function of the flow on this arc.

Consider first $t^0(i,j)$ = travel time on arc (i,j) independent of the flow on this arc. Solve the problem according to the previous algorithm γ) . The solution provides us with a flow pattern $f^0(i,j)$ which corresponds to a travel time $t^1(i,j)$.

Consider now $t^2(i,j) = \frac{t^0(i,j) + t^1(i,j)}{2}$ and solve according to the previous algorithm.

For the third iteration take $t^4(i,j) = \frac{t^2(i,j) + t^3(i,j)}{2}$.

Stop when $f^n(i,j) = f^{n-1}(i,j)$.

REFERENCES

- [1] Berge, C. and A. Ghouila-Houri, PROGRAMMES, JEUX ET RESEAUX DE TRANSPORTS, Dunod, pp. 137-142, (1962).
- [2] Dantzig, G. B., "On the Shortest Route through a Network," Management Science, Vol. 6, p. 187, (1960).
- [3] Dantzig, G. B., "All Shortest Routes in a Graph," Course Notes, I.E. 266, University of California, Berkeley.
- [4] Hu, T. C., "Multi-Commodity Network Flows," Journal of the Operations Research Society of America, Vol. 11, pp. 344-360, (1963).
- [5] Sakarovitch, M., "The Multi-Commodity Maximum Flow Problem," ORC 66-25, Operations Research Center, University of California, Berkeley, (1966).
- [6] Whitney, H., "Non-Separable and Planar Graphs," Trans. Amer. Math. Soc., Vol. 34, pp. 339-362, (1932).
- [7] Whitney, H., "Planar Graphs," Fundamenta Mathematicae, pp. 73-84, (1933).
- [8] Wollmer, R., "Removing Arcs from a Network," ORC 64-5, Operations Research Center, University of California, Berkeley, (March 1964).
- [9] Wollmer, R., "Sensitivity Analysis in Networks," ORC 65-8, Operations Research Center, University of California, Berkeley, (April 1965).

APPENDIX

APPLICATION OF THE ALGORITHM
OF SECTION II TO THE MAXIMAL FLOW PROBLEM

1. One-commodity maximum flow problem. The equivalence between a maximum flow problem in a primal planar graph $G = (N, A)$ (N is the set of nodes j) and the shortest route problem in its dual is well known (Ref. 1, 6, 7). To each arc (j_x, j_y) in the primal corresponds an arc (i_x, i_y) in the dual. To a cut in the primal corresponds a route from source to sink in the dual. To the minimal cut in the primal (which is equal to the maximal flow) corresponds the shortest route from source to sink in the dual.

Consider now a maximum flow problem on the graph G . Suppose that to each arc (j_x, j_y) corresponds a set of nonnegative numbers $c(j_x, j_y)$. Let $c^0(j_x, j_y), c^1(j_x, j_y), \dots, c^m(j_x, j_y)$ be the elements of this set with $c^0(j_x, j_y) \geq c^1(j_x, j_y) \geq \dots \geq c^m(j_x, j_y) \geq 0$ for all (j_x, j_y) . When the capacity of arc (j_x, j_y) passes from $c^0(j_x, j_y)$ to $c^\alpha(j_x, j_y)$ with $\alpha \in \{1, \dots, m\}$, we say that the capacity of arc (j_x, j_y) is reduced by $c^0(j_x, j_y) - c^\alpha(j_x, j_y)$ and that the reduced capacity is $c^\alpha(j_x, j_y)$.

Suppose s capacity-decreases can occur on the whole network. If an arc is subject to only one decrease, the reduced length is $c^1(j_x, j_y)$. If an arc is subject to more than one capacity-decrease, the successive reduced capacities are $c^1(j_x, j_y), \dots, c^m(j_x, j_y)$.

The sensitivity-analysis problem is the following: On which arcs should the s improvements occur such that the maximal flow is reduced the most?

To solve this problem, we consider the improvement problem on the dual graph $G = (N, A)$. To the arc (j_x, j_y) of G corresponds the arc (i_x, i_y) of G . To the successive capacities $c^0(j_x, j_y), \dots, c^m(j_x, j_y)$ of the primal correspond the successive arc lengths $d(i_x, i_y), \dots, b^m(i_x, i_y)$ of the dual. So the sensitivity-

A.2

analysis on the primal is equivalent to the improvement problem in the dual: On which arcs should the improvements occur such that the length of the shortest path from source to sink is reduced the most? If only one capacity-decrease can occur on each arc of the primal, we consider only one length improvement on each arc of the dual. If m capacity-decreases can occur on each arc of the primal, we consider m length improvements on each arc of the dual. The solution of the dual problem provides us with the shortest improved path from source to sink and tells us on which arcs (i_x, i_y) the improvements occurred. To the shortest path corresponds the new minimal cut in the primal and the capacity-decreases occur on the arcs (j_x, j_y) which intersect the improved dual arcs.

2. Two-commodity maximum flow problem. Consider an undirected network with two sources and two sinks. The two-commodity problem consists in maximizing the sum of the flows of two different commodities, commodity 1 being shipped from source S_1 to sink S_1^* and commodity 2 from S_2 to S_2^* .

After the following adaptation, the sensitivity-analysis problem on the two-commodity flow problem can be made in the same way as for the one-commodity problem.

T. C. Hu has shown that the maximal flow is equal to a certain minimal cut (Ref. 4). The cut here is defined as a set of arcs separating at the same time S_1 from S_1^* and S_2 from S_2^* . Hence the minimal cut can be one of the two following sorts: Either it separates S_1 and S_2 from S_1^* and S_2^* or S_1 and S_2^* from S_1^* and S_2 . Thus if we are only interested in the value of the maximum flow (and not in the flow distribution on the different arcs), we can find this value in the following way:

- Condense S_1 and S_2 into a single node W and S_1^* and S_2^* into a single node W^* . In the condensed graph G_W , consider W as source and W^* as sink. Find the minimal cut the value of which is α .
- Condense S_1 and S_2^* into a single node Z and S_1^* and S_2 into a single node Z^* . In the condensed graph G_Z , consider Z as source and Z^* as

sink. Find the minimal cut the value of which is β .

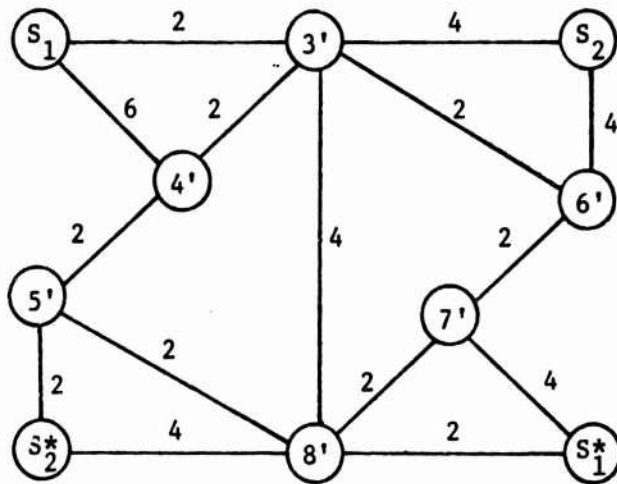
- The maximum flow for the two-commodity problem is $\min(\alpha, \beta)$.

Sensitivity-analysis. Consider the two duals G_w (corresponding to G_w) and G_z (corresponding to G_z). Make the sensitivity-analysis on both graphs. We get two values of the reduced maximum flow: α_1 and β_1 . Take $\min(\alpha_1, \beta_1)$ and keep the corresponding capacity-decreases.

Remark:

Unfortunately, this very easy method does only work in the case where the two condensed networks are planar.

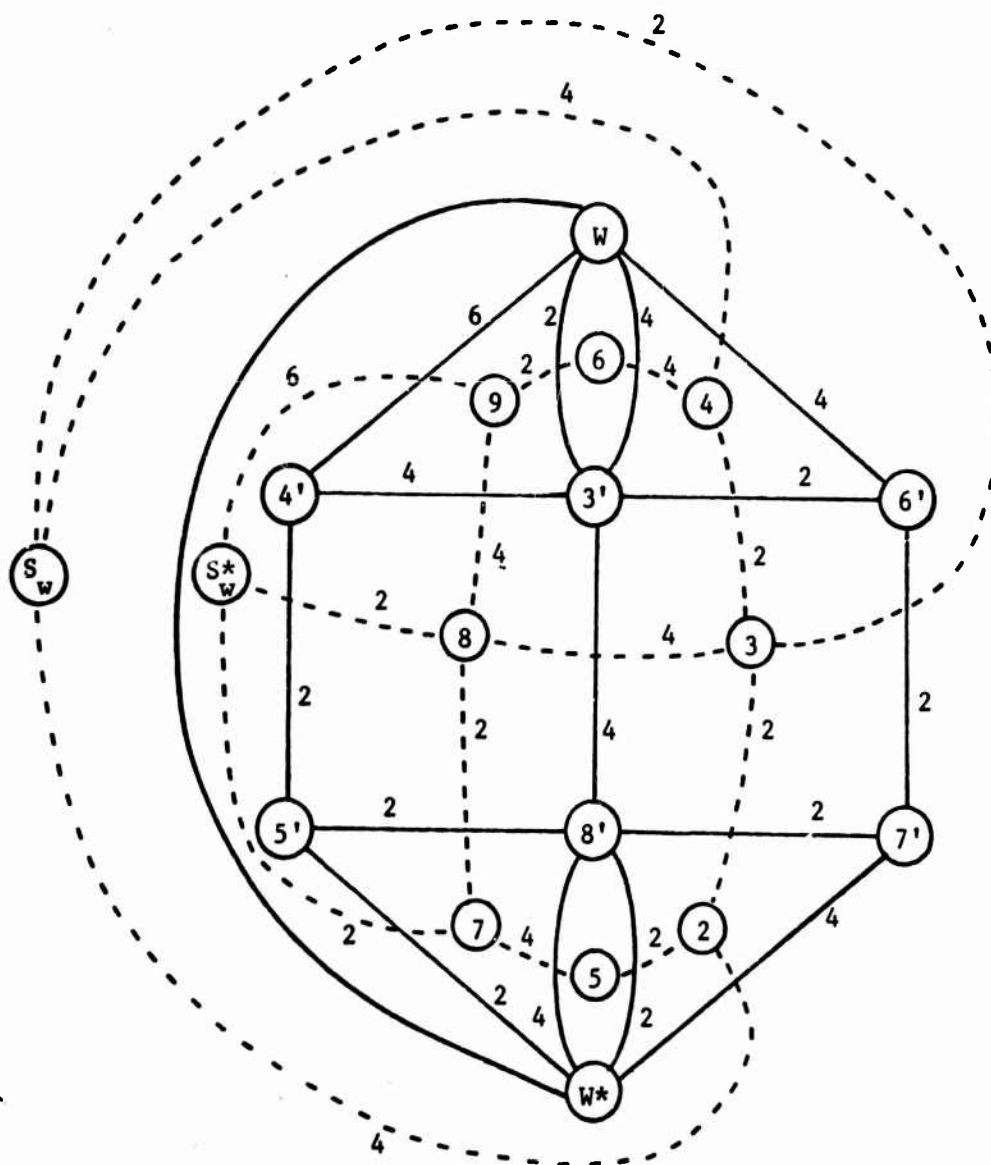
Example:



A.4

1. Put S_1 and S_2 together.

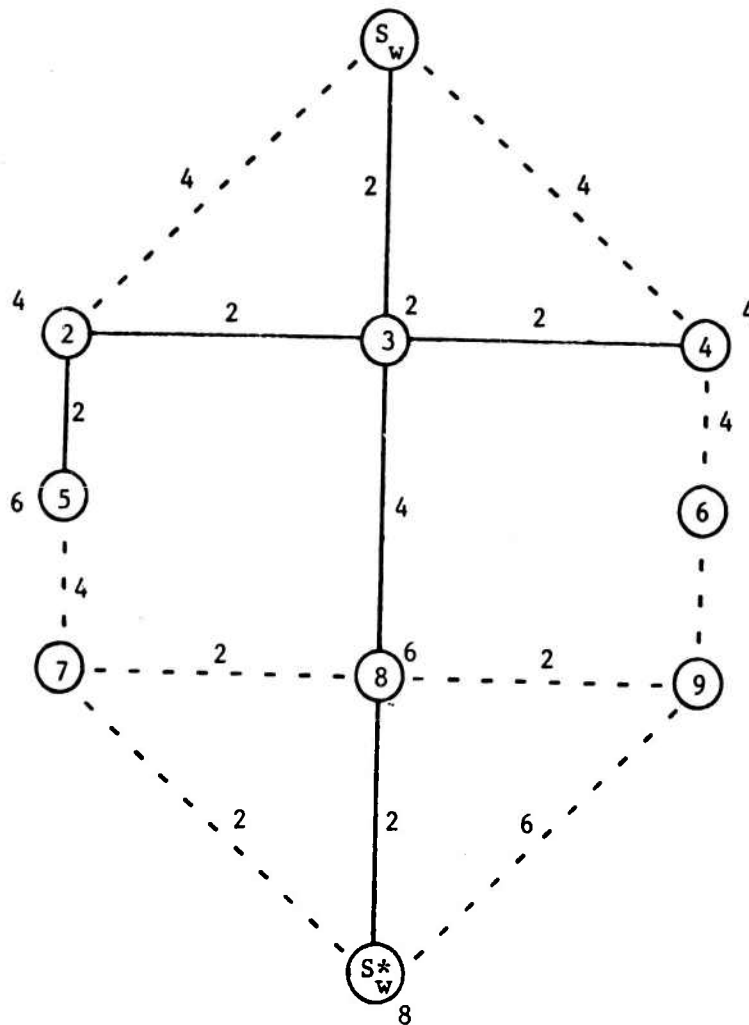
Put S_1^* and S_2^* together.



Primal Graph G_W = ———

Dual Graph G_W^* = - - - -

The dual graph is:



Length of the shortest route from source to sink = value of the minimal cut in the primal = 8.

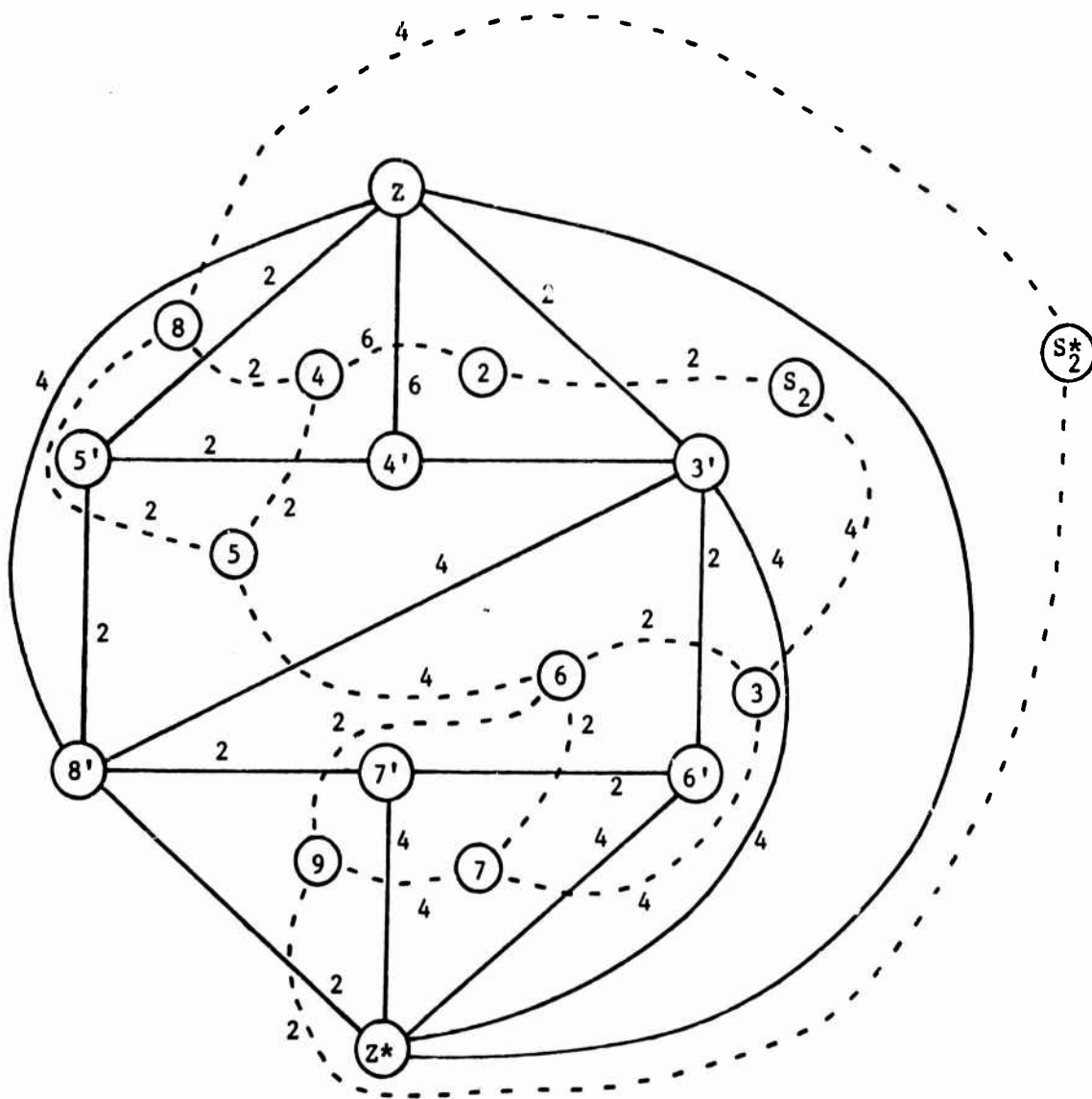
Chain: $(S_w, 3)$, $(3, 8)$, $(8, S_w^*)$.

Minimal cut: $(6', 7')$, $(3', 8')$, $(4', 5')$.

A.6

2. Put S_1 and S_2^* together.

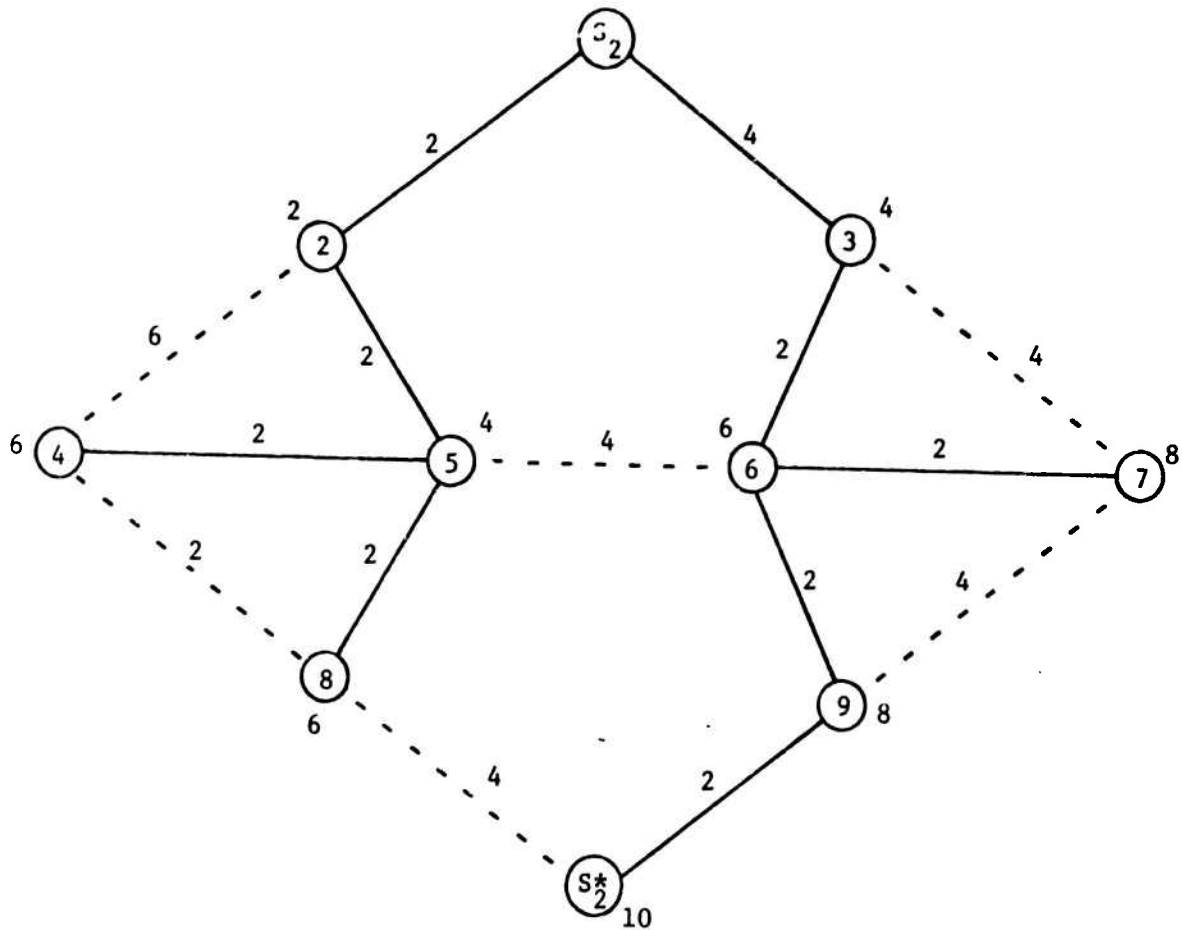
Put S_1^* and S_2 together.



Primal Graph G_z = ———

Dual Graph G_z = - - - -

The dual graph is:



Length of the shortest route from source to sink = minimal cut in the primal = 10.

Chain: $(s_2, 3)$, $(3, 6)$, $(6, 9)$, $(9, s_2^*)$.

Minimal cut: $(3', 2')$, $(3', 6')$, $(7', 8')$, $(8', 2^*)$.

Since $8 < 10$, the minimal cut is $(6', 7')$, $(3', 8')$, $(4', 5')$.

The sensitivity analysis can now be made on the two dual graphs as it has been described in Part II.

3. Multi-commodity maximum flow problem for a completely planar graph.

M. Sakarovitch (Ref. 5) defines a completely planar graph as follows: The graph G is completely planar if

- (i) The graph obtained from graph G by linking a super source to the sources of all the commodities and linking the sinks to a super sink is planar.
- (ii) The graph obtained from G by adding the arcs $(S_1, S_1^*), \dots, (S_n, S_n^*)$ is planar. (n is the number of commodities considered.)

In (5), Sakarovitch gives an algorithm to solve the multi-commodity problem in this special case:

"Begin by sending flow from S_1 to S_1^* following the general one-commodity maximum flow algorithm. Let F^1 be the quantity of flow thus sent. Define $Q_1 = (1)$. We have:

$$F^1 = c(S_1, S_1^*).$$

Then send flow from S_2 to S_2^* using the same algorithm in the network of remaining capacities; let F^2 be the amount of flow thus sent; go on sending flow from S_3 to S_3^* in the network of remaining capacities and so on."

Sakarovitch shows that this construction leads to a maximal integer multi-commodity flow and to the determination of a partition h for which

$$F^1 + \dots + F^k = \sum_{Q_i \in h} c(S_{Q_i}, S_{Q_i}^*).$$

For an n -commodity problem there exist 2^{n-1} partitions h_i . To each partition h_i corresponds the cut H_i .

The *sensitivity-analysis* can be done in the same way as for the 2-commodity case.

Consider the 2^{n-1} possible partitions h_i, A_i and B_i being the elements of the partition h_i . Condense the elements of A_i in a unique source and those of B_i in a unique sink. To the new graph G_i , corresponds the dual graph G_i . Find the shortest route from source to sink on G_i (length = α_i) which provides us with the cut H_i . $\min(\alpha_i)$ is the value of the maximal flow and the corresponding cut H_i is the minimal cut.

The sensitivity-analysis is done on the 2^{n-1} dual graphs G_i each of which provides us with a final shortest route the length of which is β_i . Take $\min(\beta_i)$ and the corresponding capacity-decreases solve the problem.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of California, Berkeley		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE TWO SENSITIVITY ANALYSIS PROBLEMS IN NETWORKS			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Research Report			
5. AUTHOR(S) (First name, middle initial, last name) DEGRAND, Joseph			
6. REPORT DATE December 1968		7a. TOTAL NO. OF PAGES 34	7b. NO. OF REFS 9
8a. CONTRACT OR GRANT NO. DA-31-174-ARO-D-331		9a. ORIGINATOR'S REPORT NUMBER(S) ORC 68-28	
b. PROJECT NO 20014501B14C		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
c.			
d.			
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES None		12. SPONSORING MILITARY ACTIVITY U.S. Army Research Office-Durham Box CM, Duke Station Durham, North Carolina 27706	
13. ABSTRACT SEE ABSTRACT.			

Unclassified
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Network Shortest Route						